Chair for Algorithms
and Data Structures
Prof. Dr. Hannah Bast
Claudius Korzen

Information Retrieval
WS 2017/2018

http://ad-wiki.informatik.uni-freiburg.de/teaching

Exercise Sheet 11

Submit until Tuesday, January 23 at **12:00pm (noon)**

Exercise 0 (20 points)

Please carefully fill out the (online) evaluation form for this course. You should have received an email with a link to the evaluation form. The evaluation is anonymous. Just confirm in your *experiences.txt* for this exercise sheet that you have completed the evaluation. However, please only confirm this after you actually did it. It's not OK to write that you did it when actually you only plan to do it. We will use the 20 points to replace the points of you worst exercise sheet. Please be honest and concrete with your feedback. We particularly value the comments in the free-text fields.

The rest of this exercise sheet is about text classification using Naive Bayes. We recommend that you proceed from the (Python) code provided on the Wiki. It already provides code for reading a set of labeled documents and constructing the corresponding document-term matrix (one row per document!) and label vector (one entry per document). It also provides specifications for Exercises 1 - 3 below as well as the unit tests.

The two main methods (*train* and *predict*) are most elegantly and most efficiently implemented using matrix-vector operations. However, if you prefer and it helps your understanding, you can also do the necessary computations using (plain and boring) for-loops.

Exercise 1 (5 points)

Write a method *train* that computes the probabilities p_c and p_{wc} for a given (training) set of documents. Use smoothing as explained in the lecture (with $\varepsilon = 0.1$) to make sure that all p_{wc} are non-zero.

[after how ignore is it naive order over please realizing to turn word]

Exercise 2 (5 points)

Write a method *predict* that computes the most likely class for each document from a given (testing) set. Use the p_c and p_{wc} learned using *train*, as explained in the lecture.

Consider the implementation advice from the lecture and add the logarithms of the appropriate probabilities, instead of multiplying the probabilities. For a more efficient prediction, you may (but do not have to) already compute the logarithms in the *train* method.

Exercise 3 (5 points)

Complete the code such that it takes the names of two files as arguments: the training data and the test data. The program should learn from the training data, then predict labels for the test data. It should output the precision, recall, and F-measure on the test data, for each class separately as well as the (unweighted) average over all classes.

Also output, for each class, the value of p_c and the 20 words with the highest p_{wc} value and which do not appear in the *stopwords.txt* provided on the Wiki.

Exercise 4 (5 points)

Run your code for the four sets of train and test data provided on the Wiki (movie genres and ratings, in two variants each). Post the average F1-scores on the table on the Wiki.

In your *experiences.txt*, briefly discuss your results. In particular, discuss the relative quality of your results (1) between the different classes (the results for some classes will be better than for others), (2) between genres and ratings, (3) between the two variants. Also discuss how much sense the top-20 words per class make and whether they are related to the precision and recall.

Please keep your discussion short. Don't write a novel. And please don't put all the numbers from the output of your code in your experiences.txt, that is TMI. Just mention the numbers which are key for your discussion (did we mention that it should be short).

Add your code to a new sub-directory *sheet-11* of your folder in the course SVN, and commit it. As usual, make sure that *compile*, *test*, and *checkstyle* run through without errors on Jenkins, so that your tutor can give you meaningful feedback also in the new year. And, of course, never forget the *experiences.txt*, where you inform us about your adventures with this exercise sheet.